

## CSE 4392 / CSE 5392 - *Smart Home Technologies*

Homework 2- Spring 2006

Due Date: March. 21 2006, 3:30 pm

### Components

- Smart Home Simulator (Running on 129.107.12.179)
- PostgreSQL Database (Running on 129.197.12.179)

Problems marked with a \* are required only for students in the graduate section (CSE 5392). They will be graded for extra credit for students of CSE 4392.

The goal of this assignment is to set up database tables for a Smart Home application and link it to data coming from a Smart Home Simulator. As part of the database, the assignment requires the setting up of triggers in order to maintain the information in the tables and to trigger alarms for intrusion and fire. For all problems you should hand in a description of your experiences and your code by submitting it to [huber@omega.uta.edu](mailto:huber@omega.uta.edu).

### Database References

PostgreSQL is a free SQL database system. Documentation about the database, including a tutorial can be found at <http://postgresql.org>. The database can also be downloaded from that site. There is also sample interface code in *C* and *Java* available.

### Smart Home Simulation

The Smart Home simulator generates an event sequence which includes a time stamp, the device identifier and the state it switched to. The general format looks as follows:

```
2006-03-01T07:00 AlClock on
```

The sensors in the environment are shown in the following map and include the following:

**Alarm Clock** [on off ]AlClock

**Lights** [on off ]Light1, Light2, Light3, Light4, Light5, Light6, Light7, Light8

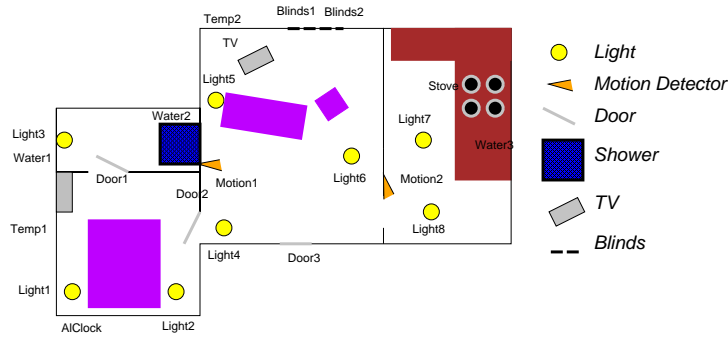
**Faucets/Shower** [on off ]Water1, Water2, Water3

**Door Crossing Sensors** [on off ]Door1, Door2, Door3

**Motion Detectors** [on off ]Motion1, Motion2

**Temperature Sensors** [60-90 ]Temp1, Temp2

**Intruder/Fire Alarm** [on off ]Alarm1, Alarm2



The simulator runs on port 8765 on 129.107.12.179. Upon connecting to it, issuing the command *run* will start the simulation. During the simulation, events are delivered one at a time and issuing *next* to the simulator causes it to issue the next event. Sending *end* to the simulator will end the simulation and disconnect. In addition to these commands, instructions to trigger a sensor or an alarm can be sent to the sensor in the form *SensorName State*, e.g. *Alarm1 on* to set off the intruder alarm.

There is a simple sample *C* and *Java* client on the web page.

## Setting up the Database and Internal Triggers

1. Create tables in your PostgreSQL database (see the script file for a possible table instantiation). You may need to edit these to add attributes needed (so read through this file carefully *\*before\** you create the tables. There is an ALTER TABLE available if you do miss anything).
2. Connect your program to the Smart Home simulator and enter sensor events into the table as they occur.
3. Create internal triggers that enter location tracking information into the tracking table according to the following scenario rules:
  - If *Motion1* is on, the person is in the living room,
  - if *Motion2* is on the person is in the Kitchen,
  - if *Motion1* is off and *Door2* is triggered the person is in the Bedroom,
  - if *Motion1* is off and *Door3* is triggered, the person has left the home.

## Alarms and External Triggers

4. Set up database triggers to enter alarm information for intruders and fire into the Alarm table using the following rules:
  - If a motion sensor or a door sensor is triggered between 10:00 am and 4:00 pm, an intruder has entered and *Alarm1* (the intruder alarm) should be set.
  - If the temperature rises to 90 in either room, *Alarm2* (the fire alarm) should be set.

- 5.\* Set up an external trigger for the alerts that turns on the appropriate alarm in the simulator (by issuing either *Alarm1 on* or *Alarm2 on* to the simulator).